

Using script

Use `"{"` DIV macro to have a tag with id

Use `"{"` Three.js macro (Now loads directly from CDN)

Use `"{"` Javascript Code Segment macro to paste the JAVA Script

Since the update of Confluence's update on Batch.js, there's a lot going on inside it.

So directly injecting javascript in page is not recommended, will try to setup Porthole+Express server example

Three.js demo code from internet

```
// Please visit : http://www.aerotwist.com/tutorials/getting-started-with-three-js/

var $container;
var camera, scene, renderer;

var pointLight = new Array();
var spheres = new Array();
var spherespeed = new Array();

init();
animate();

function init(){
    var WIDTH = 400, HEIGHT = 300;
    var CAMZ = 900;
    var VIEW_ANGLE = 45, ASPECT = WIDTH / HEIGHT, NEAR = 0.1, FAR = 10000;
    var max_spheres = 40;

    $container = $('#DisplayWebGL');
    renderer = new THREE.WebGLRenderer({ clearColor: 0x000000, clearAlpha: 1 });
    camera = new THREE.PerspectiveCamera(
        VIEW_ANGLE,
        ASPECT,
        NEAR,
        FAR);
    scene = new THREE.Scene();
    scene.add(camera);
    camera.position.z = CAMZ;
    camera.lookAt(new THREE.Vector3(0,0,0));
    renderer.setSize(WIDTH, HEIGHT);
    $container.append(renderer.domElement);

    var sphereMaterial = new THREE.MeshLambertMaterial(
        {
            color: 0xFFFFFF
        });

    var radius = 64,
        segments = 32,
        rings = 32;
    var sphere = new THREE.Mesh(
        new THREE.SphereGeometry(
            radius,
            segments,
            rings),
        sphereMaterial);
    scene.add(sphere);

    for( var i = 0; i < max_spheres; i++ ){
        spheres.push(new THREE.Mesh(
            new THREE.SphereGeometry(
                radius / (2 + i / 4),
```

```

        segments / 4,
        rings / 4
    ), sphereMaterial)
    );
    spherespeed.push( 0.5 + Math.random() * 2 );
    spheres[i].position.z = - Math.random() * 400;
    scene.add(spheres[i]);
}

pointLight.push(new THREE.PointLight(0x880000));
pointLight.push(new THREE.PointLight(0x008800));
pointLight.push(new THREE.PointLight(0x000088));
pointLight.push(new THREE.PointLight(0x888800));
pointLight.push(new THREE.PointLight(0x880088));
pointLight.push(new THREE.PointLight(0x008888));

for( var i = 0; i < pointLight.length; i++ ){
    pointLight[i].position.z = 250 + Math.floor(Math.random() * 60 - 30);
    scene.add(pointLight[i]);
}

renderer.render(scene, camera);
}

function animate(){
    requestAnimationFrame( animate );

    render();
}

function render(){
    var timer = Date.now() * 0.0005;
    for( var i = 0; i < pointLight.length; i++ ){
        timer = -timer;
        pointLight[i].position.x = Math.cos( timer * (1 + 0.3 * i) ) * 300;
        pointLight[i].position.y = Math.sin( timer * (1 + 0.3 * i) ) * 300;
    }

    for( var i = 0; i < spheres.length; i++ ){
        timer = -timer;
        spheres[i].position.x = Math.cos( timer * spherespeed[i] ) * (100 +
10 * i);
        spheres[i].position.y = Math.sin( timer * spherespeed[i] ) * (100 +
10 * i);
    }
    camera.position.x = Math.cos( timer ) * 400 ;
    camera.lookAt(new THREE.Vector3(0,0,0));

    renderer.render(scene, camera);
}

```